

an init module for determining if the timestamp data is to be collected during the operation of the application program;

a performance marker module for obtaining and storing the timestamp data for later retrieval;

an uninit module for formatting and storing the obtained timestamp data into a data file within the mass storage device that permits retrieval after the termination of the application program; and

a performance benchmark data post processing module for determining the benchmark timing from two or more timestamp data entries;

wherein the performance marker module is executed at predefined points corresponding to the permanently inserted performance markers within a plurality of processing modules within the application program.

16. (ONCE AMENDED) A method for obtaining benchmark timing for a portion of an application program execution, the application program having permanently inserted performance markers, the method comprising:

permanently inserting one or more performance markers into the application program at predefined locations within the application program corresponding to the point at which benchmark timing data is desired;

determining if benchmark timing data is to be collected at each performance marker by checking for the existence of processing modules identified by an identification key within a system registry;

if benchmark timing data is to be collected at each performance marker:

generating a benchmark data record containing the collected benchmark timing data each time the performance markers are reached;

storing the benchmark data records within a data memory block within the processing modules identified by the identification key within the system registry;

retrieving the benchmark data records from the data memory block for transfer to first data record in a Raw Data Table device once all of the run-time internal state data has been collected; and

processing the first data records stored within the Raw Data Table to generate second data records in a Processed Data Table that estimate the benchmark timing defined between two benchmark data records.

20. (ONCE AMENDED) The method according to claim 19, wherein the second data record of the Processed Data Field comprises a ResultID field, a Reboot Iteration field, a Launch Iteration field, a Marker Iteration field, a Marker Pair ID field, and a Seconds Field.

24. (ONCE AMENDED) A computer data product readable by a computing system and encoding a computer program of instructions for executing a computer process for obtaining run-time internal state data within an application program, the application program having permanently inserted performance markers, said computer process comprising:

permanently inserting one or more performance markers into the application program at predefined locations within the application program corresponding to the point at which benchmark timing data is desired;

determining if benchmark timing data is to be collected at each performance marker by checking for a processing modules identified by an identification key within a system registry;

if benchmark timing data is to be collected at each performance marker:

generating a benchmark data record containing the collected benchmark timing data each time the code markers are reached;

storing the benchmark data records within a data memory block within the processing modules identified by the identification key within the system registry;

retrieving the benchmark data records from the data memory block for transfer to first data record in a Raw Data Table device once all of the run-time internal state data has been collected; and

processing the first data records stored within the Raw Data Table to generate second data records in a Processed Data Table that estimate the benchmark timing defined between two benchmark data records;

wherein the benchmark timing generated and stored within the processed data table is determined from difference between two data entries stored within the raw data table.

At
end
Sub
B7